R E P O R T   R E S U M E S

ED 016 290 EA 001 065

AN ALGORITHM TO DETERMINE RELAT.VE IMPORTANCE OF PROJECTS.

BY- WINTERS, WILLIAM K.
NATIONAL CENTER FOR EDUCATIONAL STATISTICS (DHEW)

REPORT NUMBER TN-35 PUB DATE 31 MAY 67
EDRS PRICE MF-$0.25 HC-$1.04 24P.

DESCRIPTORS- *PROJECTS, *ALGORITHMS, *MANAGEMENT GAMES, *MANPOWER UTILIZATION, MATHEMATICAL MODELS, METHODS, COMPUTER ORIENTED PROGRAMS, DISTRICT OF COLUMBIA,

THIS NOTE PRESENTS MODELS FOR PASSING JUDGMENT ON THE IMPORTANCE OF PROJECTS. TWO APPROACHES ARE CONSIDERED--PROJECTS WITH TWO ATTRIBUTES, PRIORITY AND DEADLINE DATES, AND PROJECTS WITH K ATTRIBUTES. THE SOLUTIONS ARE ILLUSTRATED THROUGH EXAMPLES AND BY AN ALGORITHM PRESENTED IN A COMPUTER REFERENCE LANGUAGE TO SHOW HOW ONE OF THE SOLUTIONS CAN BE IMPLEMENTED ON THE COMPUTER. (HW)

ED016290

NATIONAL CENTER FOR EDUCATIONAL STATISTICS
Division of Operations Analysis

AN ALGORITHM TO DETERMINE RELATIVE IMPORTANCE OF PROJECTS

by

William K. Winters

Technical Note
Number 35

May 31, 1967

EA 001 065

OFFICE OF EDUCATION/U.S. DEPARTMENT OF HEALTH, EDUCATION, AND WELFARE

NATIONAL CENTER FOR EDUCATIONAL STATISTICS
Alexander M. Mood, Assistant Commissioner

DIVISION OF OPERATIONS ANALYSIS
David S. Stoller, Director

An Algorithm to Determine Relative Importance of Projects

Introduction

There often arises a situation in the management of the activities of an organization where it becomes a problem to keep track on a periodic basis of a large number of projects and to specify the means by which attention should be placed on the 'most important' projects at the right time. Consider the following as an illustration of the problem. Suppose that we have six projects, say project A, project B, project C, project D, project E, and project F. We have our available manpower resources working on all of these projects and further we have data of the following nature on each

Project Name	Priority Number	Time to Completion
A	7	148
B	6	153
C	1	210
D	10	165
E	9	212
F	11	192

It is not immediately obvious whether project C with a priority of 1 and a time to completion of (say) 210 (days) is more important at the present moment than, say, project A with a priority of 7 and a time to completion of 148. The reasons for this are several. First, a project's importance becomes meaningful only when it is pitted against the other projects in a set of such projects that are competing for the available manpower resources. In addition, if the criterion for passing judgment upon the importance of projects involves

only one measurable attribute, then the criterion is quite simple. One observes the magnitude of the attribute for each project and on a scale ranging from smallest to largest one associates the magnitudes with most importance to less importance respectively. The criterion becomes much more complex when an attempt is made to use more than one attribute of the projects to pass judgment on the importance of these projects.

Two approaches will be considered in this paper toward finding solutions to the problem. The case of two attributes will be considered first. Then, the more general case of an arbitrary finite number of attributes will be derived for one of the solutions. Completely worked out examples will further illustrate both solutions along with an appendix which furnishes the algorithm for one of the solutions coded in reference language ALGOL 60. There is no attempt in this paper to compare the solutions on some suitably chosen common frame of reference, rather the paper furnishes two workable and useful approaches toward handling the problem.

Priority Algorithm for the Case of Two Attributes

For the case where we consider two attributes associated with each project, the problem can be stated as follows.

Statement of the Problem. Given that we have n projects p_i $(i = 1,2,...,n)$, each with a priority number assigned to it, say y_i, what are the projects to review such that all projects can potentially be completed by their deadline date d_i, the present date being t. We want, for any t, to find the m projects such that y_i and $t - d_i$ are simultaneously minimum, for each i, where m is an arbitrarily chosen constant that represents the desired number of projects to be brought to the attention of management at time t.

Solution A: Let all projects p_i be assembled in set $P = \left\{ p_i,\ i = 1,2,\ldots,n \right\}$. For any p_i, we have the pair (y_i, z_i) where $z_i = t - d_i$. We are interested in finding that set of m pairs of (y_i, z_i) that are most important for management to be aware of at time t. Let us denote this set by

$M = \left\{ (y_j, z_j) : (y_j, z_j)\right.$ are all those pairs ranked by order of importance $\left. j = 1,2,\ldots,m \right\}$.

In order to determine this set M, we need to define the ranks of the set of n pairs, first by ranking the pairs by y_i and then by z_i.

Thus let $r_{y_i'} = i$ and $r_{z_i'} = i$ according as $y_1' < y_2' < \cdots < y_n'$ and $z_1' < z_2' < \cdots < z_n'$. Corresponding to each (y_i, z_i) we have (r_{y_i}, r_{z_i}). The final ordering of $(y_1'', z_1''), \ldots,$ (y_m'', z_m'') depends on

$$r_i = f\,(r_{y_i}, r_{z_i}).$$

The function f is selected by minimizing the distance d from the origin to any point in the xy-plane, $x > 0$ and $y > 0$. Thus

$$d = (x^2 + y^2)^{\frac{1}{2}}.$$

In substituting r_{y_i} for x and r_{z_i} for y, we obtain

$$r_i = ({r^2}_{y_i} + {r^2}_{z_i})^{\frac{1}{2}}$$

Consequently, at any time t, the m projects to review are

$$\begin{bmatrix} (y''_1,\ z''_1),\ (r_{y'_1},\ r_{z'_1}),\ r'_1,\ p'_1 \\ (y''_2,\ z''_2),\ (r_{y'_2},\ r_{z'_2}),\ r'_2,\ p'_2 \\ \vdots \\ (y''_m,\ z''_m),\ (r_{y'_m},\ r_{z'_m}),\ r'_m,\ p'_m \end{bmatrix}$$

where $r'_1 < r'_2 < \cdots < r'_m$ and $p'_j \in P$, $(j = 1,2,\ldots,m)$.

Solution B: Let us define sets $S_j^{(i)}$, $(j = 1,2;\ i = 1,2,\ldots,n)$. We will construct the sets $S_j^{(i)}$ in the following manner. $P = \left\{p_1,\ p_2,\ldots,p_n\right\}$ is an arbitrary set from which to choose. To each $p_i \in P$ we will associate $(y_i,\ z_i)$ thus we have

$$\begin{matrix} p_1,\ (y_1,\ z_1) \\ p_2,\ (y_2,\ z_2) \\ \vdots \\ p_n,\ (y_n,\ z_n) \end{matrix}$$

We will further allow y_i and z_i, for all i, to take on integer values. By ordering the pairs $(y_i,\ z_i)$ first by y_i's and then by z_i 's, we get two ordering arrays

$$
\begin{array}{ll}
p_1^{(1)},\ (y_1^{(1)},\ z_1^{(1)}) & p_1^{(2)},\ (y_1^{(2)},\ z_1^{(2)}) \\
p_2^{(1)},\ (y_2^{(1)},\ z_2^{(1)}) & p_2^{(2)},\ (y_2^{(2)},\ z_2^{(2)}) \\
\vdots & \vdots \\
p_n^{(1)},\ (y_n^{(1)},\ z_n^{(1)}) & p_n^{(2)},\ (y_n^{(2)},\ z_n^{(2)})
\end{array}
$$

where the ordering from the left array is smallest to largest moving down the array according to $y_1^{(1)} < y_2^{(1)} < \cdots < y_n^{(1)}$ and the right array is according to $z_1^{(2)} < z_2^{(2)} < \cdots < z_n^{(2)}$, $p_i^{(1)}$ and $p_i^{(2)}$ being chosen from P.

When we form the sets $S_j^{(i)}$ we do so in the following manner

$$
\begin{aligned}
S_1^{(1)} &= \{p_1^{(1)}\} \\
S_2^{(1)} &= \{p_1^{(2)}\} \\
S_1^{(2)} &= \{p_1^{(1)},\ p_2^{(1)}\} \\
S_2^{(2)} &= \{p_1^{(2)},\ p_2^{(2)}\} \\
S_1^{(3)} &= \{p_1^{(1)},\ p_2^{(1)},\ p_3^{(1)}\} \\
S_2^{(3)} &= \{p_1^{(2)},\ p_2^{(2)},\ p_3^{(2)}\} \\
&\ \vdots \\
S_1^{(n)} &= \{p_1^{(1)},\ p_2^{(1)},\ \ldots,\ p_n^{(1)}\} \\
S_2^{(n)} &= \{p_1^{(2)},\ p_2^{(2)},\ \ldots,\ p_n^{(2)}\}
\end{aligned}
$$

where $S_j^{(i)}$ is the set of elements from P chosen such that at the ith stage, the set consists of i elements of smallest magnitude ordered by their jth attribute. The algorithm that will select those p_i that occur simultaneously and are of a minimum order is the following. At each stage i, find

$$\inf S^{(i)} = S_1^{(i)} \cap S_2^{(i)}$$

which will assure us that we are selecting the p_i that occur simultaneously in $S_1^{(i)}$ and $S_2^{(i)}$. By bringing in members of P that are of minimum order in the building up process (1), we thus assure that the p_i selected at each stage are simultaneously of minimum order.

The General Priority Algorithm

For the general case when we have two or more attributes that are used to reflect the importance of projects, the problem can be stated as follows.

Statement of the Generalized Problem:

Given that we have n projects, p_i $(i = 1,2,\ldots, n)$, each with k attributes a_{ji} $(j = 1,2, \ldots, k; i = 1,2, \ldots, n)$ that are considered to reflect the importance of the p_i, at any time t we want to find the m projects such that the a_{ji} are simultaneously of minimum order on the k attributes, for each i, where m is an arbitrarily chosen constant that represents the desired number of projects to be reviewed at time t.

Solution: Let us define sets $S_j^{(i)}$ $(j = 1,2,\ldots,k;$ $i = 1,2, \ldots, n)$. We will construct the sets $S_j^{(i)}$ in the following manner. Given an arbitrary set $P = \{p_1, p_2, \ldots, p_n\}$ from which we may choose, to each $p_i \in P$ we will associate the attributes $(a_{1i}, a_{2i}, \ldots, a_{ki})$, obtaining

$$\begin{array}{l} p_1, (a_{11}, a_{21}, \ldots, a_{k1}) \\ p_2, (a_{12}, a_{22}, \ldots, a_{k2}) \\ \quad\vdots \\ p_n, (a_{1n}, a_{2n}, \ldots, a_{kn}). \end{array}$$

Consider that a_{ji}, for all i and j, take on integer values. By ordering the $(a_{1i}, a_{2i}, \ldots, a_{ki})$ by the a_{1i}'s, a_{2i}'s, ..., the a_{mi}'s, we obtain m orderings of the arrays

$$\begin{bmatrix} (p_1^{(1)}, (a_{11}^{(1)}, a_{21}^{(1)}, \ldots, a_{k1}^{(1)})), \ldots, (p_1^{(n)}, (a_{11}^{(n)}, a_{21}^{(n)}, \ldots, a_{k1}^{(n)})) \\ (p_2^{(1)}, (a_{12}^{(1)}, a_{22}^{(1)}, \ldots, a_{kn}^{(1)})), \ldots, (p_2^{(n)}, (a_{12}^{(n)}, a_{22}^{(n)}, \ldots, a_{k2}^{(n)})) \\ \vdots \qquad\qquad\qquad\qquad \vdots \\ (p_n^{(1)}, (a_{1n}^{(1)}, a_{2n}^{(1)}, \ldots, a_{kn}^{(1)})), \ldots, (p_n^{(n)}, (a_{1n}^{(n)}, a_{2n}^{(n)}, \ldots, a_{kn}^{(n)})) \end{bmatrix} \quad (1)$$

where $a_{ji}^{(q)}$ is the element that has been ordered by the qth attribute and retains the value of the jth attribute for the ith project from the set P. The first vertical array of (1) is ordered accordingly as $a_{11}^{(1)} < a_{12}^{(1)} < \cdots < a_{1n}^{(1)}$. The second array is accordingly as $a_{11}^{(2)} < a_{12}^{(2)} < \cdots < a_{1n}^{(2)}$. The corresponding orderings for the 3rd, 4th, ..., (k-1)th attributes follow until finally the kth array contains the orderings of the kth attribute accordingly as $a_{11}^{(n)} < a_{12}^{(n)} < \cdots < a_{1n}^{(n)}$.

When we form the sets $S_j^{(i)}$, we do so in the following manner.

$$\begin{aligned} S_1^{(1)} &= \{ p_1^{(1)} \} \\ S_2^{(1)} &= \{ p_1^{(2)} \} \\ &\vdots \\ S_k^{(1)} &= \{ p_1^{(k)} \} \\ S_1^{(2)} &= \{ p_1^{(1)}, p_2^{(1)} \} \\ S_2^{(2)} &= \{ p_1^{(2)}, p_2^{(2)} \} \qquad (2) \\ &\vdots \end{aligned}$$

$$S_k^{(2)} = \{ p_1^{(k)}, p_2^{(k)} \}$$

$$S_1^{(n)} = \{ p_1^{(1)}, p_2^{(1)}, \ldots, p_n^{(1)} \}$$

$$S_2^{(n)} = \{ p_1^{(2)}, p_2^{(2)}, \ldots, p_n^{(2)} \}$$

$$\vdots$$

$$S_k^{(n)} = \{ p_1^{(k)}, p_2^{(k)}, \ldots p_n^{(k)} \}$$

The algorithm used to select those p_i that occur simultaneously and are of minimum order is the following. At each stage i when the above sets (2) are formed, find

$$\inf S^{(i)} = \bigcap_{j=1}^{k} S_j^{(i)}$$

will assure us that we are selecting the p_i that occur simultaneously in each ordered array $S_1^{(i)}, S_2^{(i)}, \ldots, S_k^{(i)}$. By bringing in members of P that are of minimum order in the "building up" process of (2), we guarantee that the p_i selected at each stage are of simultaneous minimum order.

An Example with Two Attributes using Solution A

Consider that we have the following projects with the two attributes "priority number" and "time until completion".

Priority Number	Time Until Completion	Project
20	231	p_1
3	242	p_2
10	151	p_3
5	50	p_4
18	20	p_5

Priority Number	Time Until Completion	Project
1	250	p_6
2	30	p_7
4	180	p_8
6	201	p_9
12	75	p_{10}

Say that I want to know: what are the five most important projects that, as a manager in control of these projects, I wish to "crack the whip on" or "ride herd on" today? The answer is p_7, p_4, p_8, p_3, p_{10}, in that order. The reason is: the priority and the time to completion of each of the above are both simultaneously at a minimum for the above (ordered) five projects.

Solution: Ranking each of the projects by "priority number" we have:

Priority Rank	Priority Number	Time Until Completion	Project
1	1	250	p_6
2	2	30	p_7
3	3	242	p_2
4	4	180	p_8
5	5	50	p_4
6	6	201	p_9
7	10	151	p_3
8	12	75	p_{10}
9	18	20	p_5
10	20	231	p_1

Ranking each of the projects by "time until completion" we have:

Time Until Completion Rank	Priority Number	Time Until Completion	Project
1	18	20	p_5
2	2	30	p_7
3	5	50	p_4
4	12	75	p_{10}
5	10	151	p_3
6	4	180	p_8
7	6	201	p_9
8	20	231	p_1
9	3	242	p_2
10	1	250	p_6

Next associating both the priority rank and the time to completion rank with each of the projects and putting the project numbers in order we have:

Priority Number	Time Until Completion	Project	Priority Rank	Time to Completion Rank	r^2
20	231	p_1	10	8	164
3	242	p_2	3	9	90
10	151	p_3	7	5	74
5	50	p_4	5	3	34
18	20	p_5	9	1	82
1	250	p_6	1	10	101
2	30	p_7	2	2	8
4	180	p_8	4	6	52
6	201	p_9	6	7	85
12	75	p_{10}	8	4	80

Finally, ranking the projects by their measure of importance r^2 we have:

	1	2	3	4	5	6	7	8	9	10
Project	p_7	p_4	p_8	p_3	p_{10}	p_5	p_9	p_2	p_6	p_1
Measure of Importance r^2	8	34	52	74	80	82	85	90	101	164

Thus the first five "most important" project to review are $(p_7, p_4, p_8, p_3, p_{10})$.

An Example with Three Attributes using Solution B

In this example consider that we have three attributes related to each project: priority number 1, priority number 2, and time to completion. The numerical values for each attribute are listed below.

Priority No. 1	Priority No. 2	Time to Completion	Project
2	4	52	p_1
18	14	79	p_2
7	10	203	p_3
10	3	45	p_4
8	7	68	p_5
4	12	93	p_6
5	9	112	p_7
15	1	126	p_8
1	5	54	p_9
12	21	88	p_{10}
23	11	156	p_{11}
6	8	29	p_{12}
20	18	169	p_{13}

Priority No. 1	Priority No. 2	Time to Completion	Project
25	2	90	p_{14}
19	24	132	p_{15}

By ordering the projects first by priority one we obtain the following.

Priority No. 1	Priority No. 2	Time Until Completion	Project
1	5	54	p_9
2	4	52	p_1
4	12	93	p_6
5	9	112	p_7
6	8	29	p_{12}
7	10	203	p_3
8	7	68	p_5
10	3	45	p_4
12	21	88	p_{10}
15	1	126	p_8
18	14	79	p_2
19	24	132	p_{15}
20	18	169	p_{13}
23	11	156	p_{11}
25	2	90	p_{14}

Next ordering the projects on priority two we have:

Priority No. 1	Priority No. 2	Time Until Completion	Project
15	1	126	p_8
25	2	90	p_{14}
10	3	45	p_4
2	4	52	p_1
1	5	54	p_9
8	7	68	p_5
6	8	29	p_{12}
5	9	112	p_7
7	10	203	p_3
23	11	156	p_{11}
4	12	93	p_6
18	14	79	p_2
20	18	169	p_{13}
12	21	88	p_{10}
19	24	132	p_{15}

We get the following by ordering the projects on time to completion.

Priority No. 1	Priority No. 2	Time Until Completion	Project
6	8	29	p_{12}
10	3	45	p_4
2	4	52	p_1
1	5	54	p_9
8	7	68	p_5
18	14	79	p_2
12	21	88	p_{10}
25	2	90	p_{14}
4	12	93	p_6
5	9	112	p_7
15	1	126	p_8
19	24	132	p_{15}
23	11	156	p_{11}
20	18	169	p_{13}
7	10	203	p_3

We next present in the following an ordered list of projects by each attribute.

Projects Ordered by Priority No. 1	Projects Ordered by Priority No. 2	Projects Ordered According to Time to Completion
p_9	p_8	p_{12}
p_1	p_{14}	p_4
p_6	p_4	p_1
p_7	p_1	p_9
p_{12}	p_9	p_5
p_3	p_5	p_2
p_5	p_{12}	p_{10}
p_4	p_7	p_{14}
p_{10}	p_3	p_6
p_8	p_{11}	p_7
p_2	p_6	p_8
p_{15}	p_2	p_{15}
p_{13}	p_{13}	p_{11}
p_{11}	p_{10}	p_{13}
p_{14}	p_{15}	p_3

Now, by bringing in elements from each ordered list to build up sets we have the following.

$$
\begin{aligned}
s_1^{(1)} &= \{p_9\} \\
s_2^{(1)} &= \{p_8\} \\
s_3^{(1)} &= \{p_{12}\} \\
\inf s^{(1)} &= \emptyset \\
s_1^{(2)} &= \{p_9, p_1\} \\
s_2^{(2)} &= \{p_8, p_{14}\} \\
s_3^{(2)} &= \{p_{12}, p_4\} \\
\inf s^{(2)} &= \emptyset \\
s_1^{(3)} &= \{p_9, p_1, p_6\} \\
s_2^{(3)} &= \{p_8, p_{14}, p_4\} \\
s_3^{(3)} &= \{p_{12}, p_4, p_1\} \\
\inf s^{(3)} &= \emptyset \\
s_1^{(4)} &= \{p_9, p_1, p_6, p_7\} \\
s_2^{(4)} &= \{p_8, p_{14}, p_4, p_1\} \\
s_3^{(4)} &= \{p_{12}, p_4, p_1, p_9\} \\
\inf s^{(4)} &= \{p_1\} \\
s_1^{(5)} &= \{p_9, p_1, p_6, p_7, p_{12}\}
\end{aligned}
$$

$$S_2^{(5)} = \{p_8, p_{14}, p_4, p_1, p_9\}$$

$$S_3^{(5)} = \{p_{12}, p_4, p_1, p_9, p_5\}$$

$$\inf S^{(5)} = \{p_1, p_9\}$$

$$S_1^{(6)} = \{p_9, p_1, p_6, p_7, p_{12}, p_3\}$$

$$S_2^{(6)} = \{p_8, p_{14}, p_4, p_1, p_9, p_5\}$$

$$S_3^{(6)} = \{p_{12}, p_4, p_1, p_9, p_5, p_2\}$$

$$\inf S^{(6)} = \{p_1, p_9\}$$

$$S_1^{(7)} = \{p_9, p_1, p_6, p_7, p_{12}, p_3, p_5\}$$

$$S_2^{(7)} = \{p_8, p_{14}, p_4, p_1, p_9, p_5, p_{12}\}$$

$$S_3^{(7)} = \{p_{12}, p_4, p_1, p_9, p_5, p_2, p_{10}\}$$

$$\inf S^{(7)} = \{p_1, p_9, p_{12}, p_5\}$$

$$S_1^{(8)} = \{p_9, p_1, p_6, p_7, p_{12}, p_3, p_5, p_4\}$$

$$S_2^{(8)} = \{p_8, p_{14}, p_4, p_1, p_9, p_5, p_{12}, p_7\}$$

$$S_3^{(8)} = \{p_{12}, p_4, p_1, p_9, p_5, p_2, p_{10}, p_{14}\}$$

$$\inf S^{(8)} = \{p_1, p_9, p_{12}, p_5, p_4\}$$

Thus, if we were looking for the five most important projects to review based upon the three attributes and given the 15 projects under consideration we find

$\{p_1, p_9, p_{12}, p_5, p_4\}$ in that order.

Summary and Conclusions

This paper has in objective terms formulated a problem which is faced quite often by service bureau type organizations. Two approaches to the solution of the problem have been formulated. The solutions have been illustrated through examples and an algorithm has been presented in a computer reference language to show how one of the solutions can be implemented on the computer.

In solution A the case of two attributes has been presented. The approach could be extended and generalized by considering a more elaborate distance function in n-dimensional space. Further the distance function could include the provision for weighting the attributes in order to place more emphasis on some attributes that according to good management judgment may deserve such treatment. It is very apparent that many variations of the solutions presented could be devised to fit special applications, however, in most situations the variations involve subjective management judgment that would most likely come from using and experimenting with the algorithm.

APPENDIX

ALGOL PROGRAM FOR SOLUTION A

```
procedure priority(p,a,q,k,m,n); value k,m,n; array p,q,a;
integer k,m,n;
comment This algorithm will for n projects p1,p2,...,pn and
  their associated k attributes a1i,a2i, ... ,aki find the
  m 'most important' projects using solution A and will
  leave them ordered in the array q;
begin array b[1:k,1:n], d[1:k,1:n], e[1:k,1:n], c[1:n];
  integer i,j,L; real temporary,smallest,index;
  for j:=1 step 1 until k do
  begin
    for L:=1 step 1 until n do
    begin smallest:=a[j,1]; index :=1;
      for i:=1 step 1 until n-1 do
      begin
        if smallest ≠ 0 then
        begin
          if a[j,i+1]≠ 0 then
          begin
            if a [j,i+1] < smallest then
            begin
              index :=i+1; smallest := a[j,i+1]; go to A1;
            end else go to A1
          end else go to A1
        end else
```

```
      smallest := a[j,i];
Al: end i loop;
    b[j,L] := index; a[j,index] :=0; d[j,L] :=L; else
  end L loop;
end j loop;
comment associate attribute ranks with each project and
        compute the new priority rank for each;
for L:=1 step 1 until n do
begin
  for j:=1 step 1 until k do
  begin
    smallest := b[j,1]; index :=1;
    for i:=1 step 1 until n do
    begin
      if smallest ≠ 0 then
      begin
        if b j,i+1 ≠ 0 then
        begin
          if b[j,i+1] < smallest then
          begin
            index := i+1; smallest := b[j,i+1]; go to Bl;
          end else go to Bl
        end else  go to Bl
      end else
```

```
        smallest := b[j,i];
B1: end i loop;
      e[j,L]:=d[j,index] ↑ 2;
   end j loop;
   for j:=1 step 1 until k do temporary := temporary + e[j,L];
   c[L]:= sqrt (temporary);
end L loop;
comment order the projects by their new priority ranks and
        store most important projects in q array;
for i:=1 step 1 until n do
begin
   smallest :=c[1]; index :=1;
   for L:=1 step 1 until n-1 do
   begin
     if smallest ≠ 0 then
     begin
       if c[L+1]≠ 0 then
       begin
         if c[L+1] < smallest then
         begin
           index := L+1; smallest := c[L+1]; go to C1;
         end else go to C1
       end else go to C1
     end else
     smallest := c[L]; go to C1
```

```
        q[i] := p[L]
Cl: end L loop;
  end i loop;
end priority procedure;
```